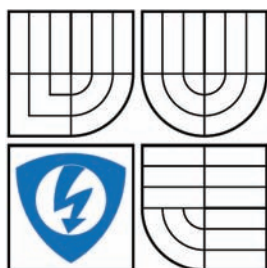


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

VIZUALIZACE MANUÁLNÍHO OVLÁDÁNÍ MODELU MANIPULÁTORU

VISUALISATION OF MANUAL CONTROL OF MANIPULATOR MODEL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

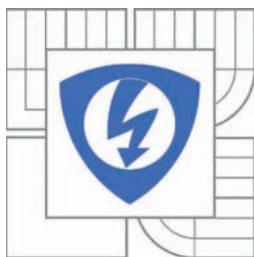
AUTOR PRÁCE
AUTHOR

PETR CHLAD

VEDOUcí PRÁCE
SUPERVISOR

ING. RADEK ŠTOHL, PH.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Petr Chlad

Ročník: 3

ID: 106485

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Vizualizace manuálního ovládání modelu manipulátoru

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s instrumentací firmy Rockwell Automation pro model manipulátoru a vizualizací pomocí prostředí firmy Rockwell Automation.
2. Vytvořte vizualizaci manuálního ovládání 3 os modelu manipulátoru.
3. Vytvořte ukázkovou laboratorní úlohu.

DOPORUČENÁ LITERATURA:

Kinetix 2000 Multi-axis Servo Drive. (User Manual) Milwaukee: Rockwell Automation, Inc. 2007. 226 s.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Radek Štohl, Ph.D.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Cílem bakalářské práce bylo vytvořit vizualizaci manuálního ovládání modelu manipulátoru. Manipulátor se skládá ze servomotorů a měničů. Je řízen programovatelným automatem GuardLogix. Vizualizace je zobrazena na barevném dotykovém panelu PanelView Plus 700.

KLÍČOVÁ SLOVA

Vizualizace, Programovatelný automat, Ovládání, Manipulátor

ABSTRACT

Purpose of this bachelor's thesis was to create a visualization of the manual control of the manipulator model. Manipulator consists of servomotors and converters. It is controled by programmable logic controller GuardLogix. Visualization is displayed on a color touch panel PanelView Plus 700.

KEYWORDS

Visualization, Programmable Logic Controller, Control, Manipulator

CHLAD P. *Vizualizace manuálního ovládání modelu manipulátoru*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. Počet stran 45. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Vizualizace manuálního ovládání modelu manipulátoru jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: 31. května 2010

.....

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlvi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 31. května 2010

.....

podpis autora

OBSAH

1 Úvod	9
2 Řešení bakalářské práce	10
2.1 Úvod	10
2.2 Model manipulátoru	11
2.2.1 AC servomotory	11
2.2.2 Systém Kinetix 2000	12
2.2.3 Limitní spínače a snímače polohy	13
2.3 PLC GuardLogix	15
2.3.1 Šasi	15
2.3.2 Zdroj	16
2.3.3 Procesory	17
2.3.4 Modul EtherNet/IP	18
2.3.5 Modul SERCOS Interface	18
2.4 PanelView Plus 700	20
2.5 RS Linx	21
2.6 RS Logix 5000	22
2.6.1 Nastavení	22
2.6.2 Proměnné	23
2.6.3 Instrukce	25
2.6.4 Program	27
2.7 RS View Studio (Factory Talk View Studio)	30
2.7.1 Nastavení	30
2.7.2 Objekty	31
2.7.3 Vizualizace	32
3 Závěr	39
Reference	40
Seznam symbolů, veličin a zkratk	43
Seznam příloh	44
A Obsah CD s přílohami	45

SEZNAM OBRÁZKŮ

2.1	Schéma zapojení	10
2.2	Model manipulátoru	11
2.3	AC servomotor řady MP [13]	12
2.4	Kinetix 2000 v rozváděči manipulátoru	12
2.5	Indukčnostní senzor Pulsotronic, limitní spínač RA řady 440P [15][16] . .	13
2.6	PLC třídy GuardLogix	15
2.7	Šasi 1756-A7/A [8]	16
2.8	AC napájecí zdroj	16
2.9	GuardLogix procesor a Safety Partner procesor	17
2.10	Modul EtherNet/IP	18
2.11	Modul SERCOS Interface	19
2.12	PanelView Plus 700	20
2.13	Vývojový diagram programu	29
2.14	Prostředí RS View Studio [11]	30
2.15	Konfigurace os	33
2.16	Konfigurace os - chyba	33
2.17	Start inicializace	33
2.18	Probíhající inicializace	33
2.19	Ovládání manipulátoru - stav po inicializaci	36
2.20	Historie poloh os	36
2.21	Překročení software limitu	37
2.22	Displej Setting	38
2.23	Vkládání hodnot	38

SEZNAM TABULEK

2.1	Parametry použitých převodníků	13
2.2	Moduly PLC a připojená zařízení	21
2.3	Společné proměnné	23
2.4	Proměnné pro osu	24
2.5	Stavy osy v Multistate indicator	34

1 ÚVOD

Hlavním cílem bakalářské práce bylo navrhnout koncepci vizualizace manuálního ovládání tří os modelu manipulátoru. Poté vytvořit program pro programovatelný automat GuardLogix, na kterém bude možné vizualizaci otestovat.

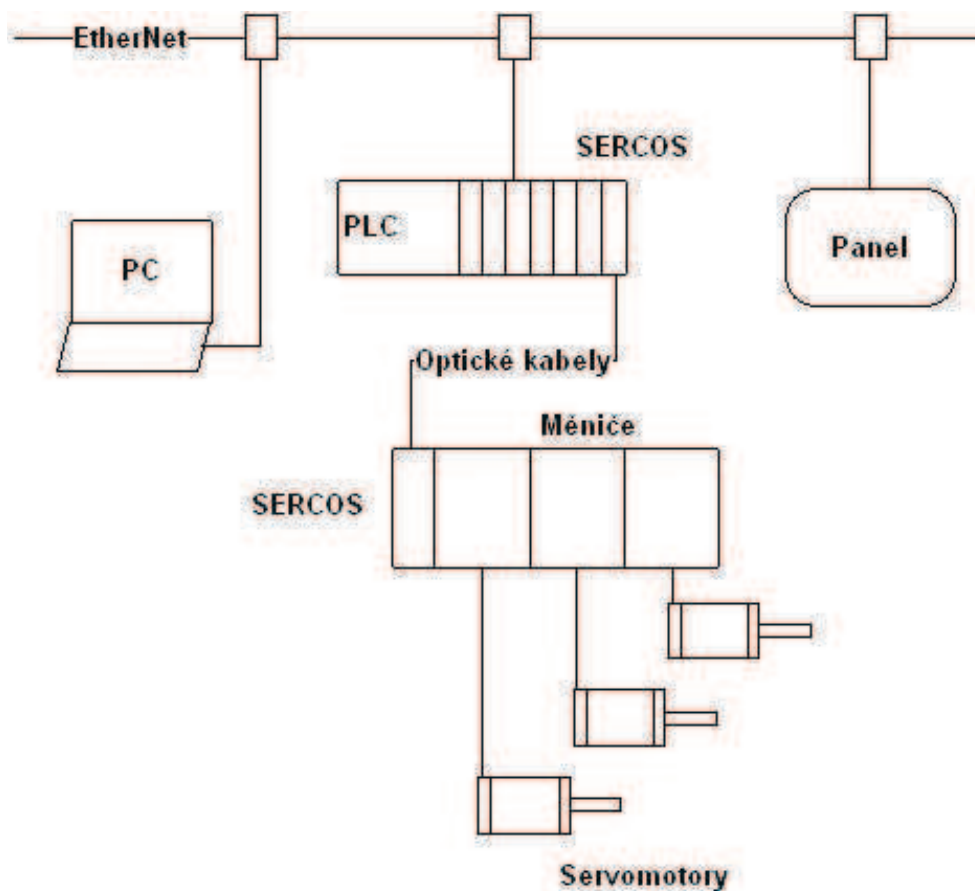
Převážná většina komponent je od společnosti Rockwell Automation. Nejprve se proto musíme seznámit s konkrétní instrumentací. V kapitole 2.2 jsou představeny jednotlivé části modelu manipulátoru. V kapitole 2.3 je popsán programovatelný automat a jeho části. Kapitola 2.4 přibližuje grafický panel pro zobrazení vizualizace. Programové vybavení je od firmy Rockwell Software, která je součástí Rockwell Automation. V kapitole 2.5 je popsán program RS Linx. Kapitola 2.6 ukazuje naprogramování samotného PLC v programu RS Logix 5000, nastavení, vytvoření proměnných, popis použitých instrukcí a samotný program. Hlavní cíl projektu je v kapitole 2.7, kde je výsledný vzhled vizualizace, funkce a postup jejího vzniku.

2 ŘEŠENÍ BAKALÁŘSKÉ PRÁCE

2.1 Úvod

Hardware manipulátoru je složen převážně z komponent americké společnosti Rockwell Automation, která sdružuje firmy Allen-Bradley a Rockwell Software. Dle funkce jednotlivých částí lze hardware rozdělit na tři části: model manipulátoru, řídicí PLC a vizualizaci.

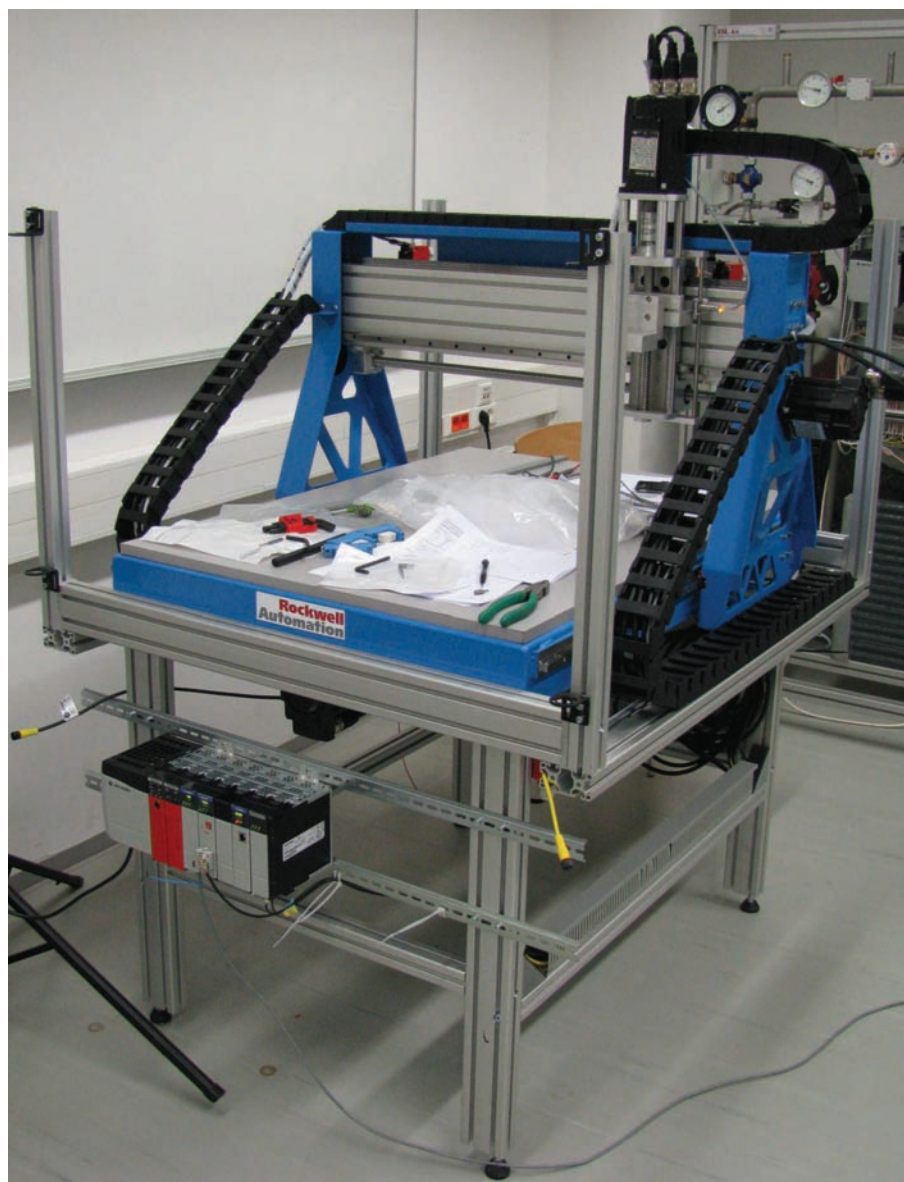
Model manipulátoru se skládá z AC servomotorů, převodníků a konstrukce samotného manipulátoru, osazené dvěma druhy snímačů. Řídicí člen je programovatelný automat třídy GuardLogix. Jako vizualizační prostředek je dotykový displej PanelView Plus 700. K programování byl použit běžný stolní počítač. Software RS Logix 5000 slouží přímo k programování funkce PLC. V programu RS View Studio (Factory Talk View Studio) se vytváří vizualizace.



Obrázek 2.1: Schéma zapojení

2.2 Model manipulátoru

Na konstrukci manipulátoru jsou upevněny motory a snímače pro každou osu. Na přední straně je uchyceno řídicí PLC a na zadní rozváděč s měniči. V pracovním prostoru je deska, kterou lze využít např. na přilepení papíru pro kreslení.

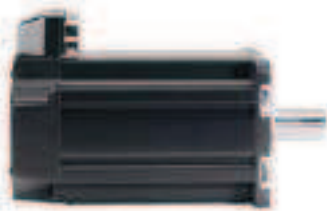


Obrázek 2.2: Model manipulátoru

2.2.1 AC servomotory

Motory Rockwell Automation řady MP [13] s nízkou setrvačností jsou bezkomutátorové výkonné servomotory. Využívají konstrukce, která redukuje velikost motoru a sou-

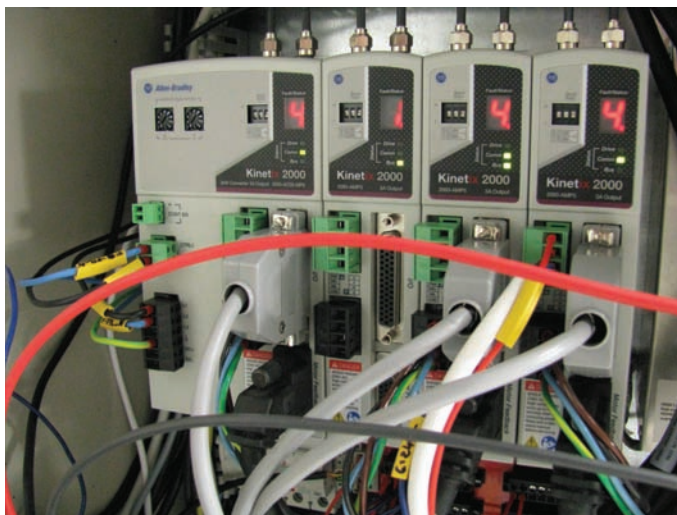
časně výrazně zvyšuje točivý moment. Tyto motory umožňují široký rozsah točivých momentů, rychlost do 5000 otáček za minutu a regulaci s vysokým rozlišením (do 2 milionů kroků na otáčku). Jsou dostupné s vinutím pro 230V a 460V. Používají se např. na balení, zpracování, manipulaci s materiálem, tváření kovů, v automobilovém průmyslu atd.



Obrázek 2.3: AC servomotor řady MP [13]

V této práci byly použity dva motory MPL-A310F-HK22AA a jeden MPL-A310F-HK24AA s brzdou 24VDC. Mají nominální rychlost 3000 otáček za minutu, rozlišení 2000 kroků na otáčku a vinutí pro 230V.

2.2.2 Systém Kinetix 2000



Obrázek 2.4: Kinetix 2000 v rozváděči manipulátoru

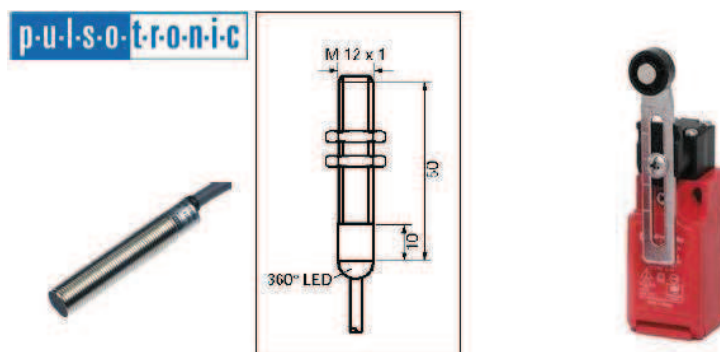
Systémy Kinetix [13] patří mezi víceosé systémy Motion řízení. Byly navrženy pro řízení servomotorů. Využívají řídicí platformu Logix pro diskrétní řízení a řízení procesů. ControlLogix zároveň umožňuje řízení kinematiky robotů.

Systém Kinetix 2000 se skládá z IAM převodníku (Integrated Axis Module) a z několika AM převodníků (Axis Module), pro každý motor jeden převodník. AM převodník nelze použít bez IAM převodníku. Komunikace s PLC je zajištěna prostřednictvím optických kabelů a modulu Sercos Interface, který zajišťuje komplexní, víceosé a synchronizované řízení až osmi os najednou.

Tabulka 2.1: Parametry použitých převodníků

Označení	2093-AC05-MP5 & 2093-AMP5
Popis	IAM převodník & AM převodníky
Nepřetržitý výst. I (EF.)	3 A
Nepřetržitý výst. I (0-špička)	4,24 A
Jmenovitý špičkový výst. I (EF.)	9 A
Jmenovitý špičkový výst. I (0-špička)	12,73 A
Nepřetržitý výst. výkon (jmenovitý)	0,9 kW

2.2.3 Limitní spínače a snímače polohy



Obrázek 2.5: Indukční senzor Pulsotronic, limitní spínač RA řady 440P [15][16]

Pro bezpečnost jsou na každé ose manipulátoru nainstalovány dva druhy senzorů. Bezpečnostní spínače vyhodnocují fyzické překročení rozsahu os. Na kalibraci polohy a definování nulové (home) pozice na osách se využívají indukční senzory.

Jako bezpečnostní spínače jsou použity spínače RA řady 440P-C 22mm [16]. Jedná se o spínač s plastovým krytem a nastavitelnou páčkou, který obsahuje jeden bezpečnostní kontakt se západkou a pomocný kontakt se západkou. Připojení se provádí pomocí konektoru s označením M20. Krytí celého snímače je IP 66.

Snímače home polohy jsou senzory Pulsotronic [15]. Jsou to indukčnostní senzory PNP ve válcovitém pouzdře (mosazné, poniklované) s metrickým závitem M12 s dosahem 4mm. Napájecí napětí je v rozsahu 10 - 35 V DC, spínací frekvence 2500 Hz, krytí IP 67. Signalizace stavu je pomocí LED diody a připojení pomocí kabelu 3x0,34.

2.3 PLC GuardLogix



Obrázek 2.6: PLC třídy GuardLogix

PLC automat třídy GuardLogix [1][3][8] je speciální verze systému ControlLogix, která se používá v bezpečných aplikacích. Spočívá ve spojení standardní a bezpečnostní funkce do jednoho projektu. Hlavní rozdíl od standardních PLC je v použití dvou procesorů, kde jeden pracuje jako běžný automat a druhý vykonává navíc ještě bezpečnostní (safety) funkce. Tento systém je certifikovaný pro použití v bezpečných aplikacích splňujících SIL3 úroveň normy IEC 61508 a IEC 61511 a bezpečnostní kategorii CAT4 normy EN 945-1.

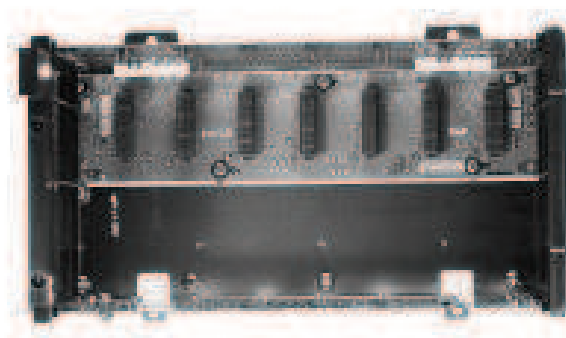
SIL (Safety Integrity Level)[2] vyjadřuje míru schopnosti produktu snížit riziko nebezpečné chyby (události), která může nastat. Definuje výslednou schopnost řídicího systému pracovat správně z hlediska bezpečnosti. SIL2 a SIL3 jsou běžné úrovně pro stroje a procesní bezpečné aplikace. GuardLogix může být použit v aplikacích splňujících tyto úrovně.

GuardLogix PLC mohou komunikovat prostřednictvím EtherNet/IP, ControlNet, DeviceNet, DH+, Remote I/O a RS-232-C (DF1/DH-485 protokol) atd. Pro zavedení komunikace stačí přidat do šasi příslušný modul požadovaného rozhraní.

2.3.1 Šasi

GuardLogix je modulární systém vyžadující šasi řady 1756 I/O [8]. Jakýkoli modul této řady je možno zapojit na libovolné místo s výjimkou zdroje, který se umísťuje vždy na levý okraj. Zadní stěna (backplane) poskytuje napájení a vysokorychlostní komunikaci mezi moduly. Všechna šasi jsou navržena pouze pro horizontální umístění. Vodicí lišta

a systém Snap-in umožňují snadné a bezpečné uchycení modulů řady 1756. Šasi lze volit v provedení se 4, 7, 10, 13 nebo 17 sloty. V tomto případě je použito šasi 1756-A7/A se 7 sloty.



Obrázek 2.7: Šasi 1756-A7/A [8]

2.3.2 Zdroj



Obrázek 2.8: AC napájecí zdroj

ControlLogix zdroje [8] se umísťují do šasi řady 1756. Standardní zdroje se umísťují výhradně na levý okraj šasi, kde se zasunou přímo na zadní lištu. Zdroj přímo napájí zadní stěnu (backplane). Poskytuje napětí 1,2V, 3,3V, 5,1V a 24V.

U PLC pro manipulátor je použit zdroj s označením 1756-PA72/C. Je napájen 220V AC. Maximální příkon je 100W a výkon 75W.

2.3.3 Procesory



Obrázek 2.9: GuardLogix procesor a Safety Partner procesor

Jak již bylo zmíněno, PLC GuardLogix [4] je ControlLogix systém, který se používá v bezpečných aplikacích. Je založen na principu dvou paralelně běžících procesorů. Hlavní výhodou tohoto systému je soustředění standardní a bezpečnostní funkce do jednoho celku.

Primární procesor vykonává standardní řídicí funkce a komunikuje se Safety Partner procesorem. Primární procesor se skládá z centrálního procesoru, I/O rozhraní a paměti. Safety Partner procesor je část systému, která je automaticky nakonfigurována a nevyžaduje samostatné nastavení, konfiguraci a stahování dat. Provádí nadřazené bezpečnostní funkce v systému.

Jako primární automat lze volit některý z řady 1756-L6xS a druhý bezpečnostní je 1756-LSP. Při programování Safety programu a úloh lze využít pouze Ladder diagram. Standardní úlohy lze programovat obvyklými jazyky pro PLC (LD-Ladder diagram, FBD-Function Block Diagram, IL-Instruction List, SFC-Sequential Function Chart, ST-Structured text). Po otestování projektu a jeho dokončení (po finální validaci) uživatel bezpečnostní úlohu uzamkne. Když je bezpečnostní paměť uzamčena a chráněna, tak nemůže být změněna bezpečnostní logika včetně konfigurace a bezpečnostní funkce pracují v úrovni SIL3 a bezpečnostní kategorii 4. Ve standardní části PLC všechny funkce pracují jako na regulérním řídicím systému Logix. Je tedy možná On-line editace, nastavení a jiné činnosti jsou povoleny. Díky integraci bezpečnostního prvku do standardního řídicího systému je možné číst bezpečnostní paměť pomocí externích zařízení (jednotky HMI apod.).

Procesor může být umístěn do libovolné pozice v ControlLogix šasi. Výjimku tvoří Safety Partner, který musí být umístěn výhradně vpravo od primárního procesoru. Do

jednoho šasi může být umístěno více procesorů. Více procesorů v jednom šasi může mezi sebou komunikovat bez sítě prostřednictvím backplane. Přitom ale pracují samostatně a nejsou na sobě nijak závislí. Použitím CIP Safety protokolu na síti Ethernet lze spojit více procesorů GuardLogix. To umožňuje, aby více jednotek GuardLogix sdílelo bezpečnostní data pro vzájemnou komunikaci mezi různými stroji, technologickými celky nebo oblastmi.

2.3.4 Modul EtherNet/IP



Obrázek 2.10: Modul EtherNet/IP

EtherNet/IP (Industrial protocol) protokol je průmyslový síťový standard. Jedná se o průmyslovou verzi Ethernetu. Je jím propojeno PC s PLC a panelem. Využíván byl zejména pro nahrávání řídicího programu do PLC nebo vizualizace do panelu. Během samotného ovládání manipulátoru samozřejmě zajišťuje předávání veškerých povelů a signálů mezi panelem a PLC.

V PLC je konkrétně modul 1756-ENBT [14]. Tento modul je schopen komunikovat rychlostí 10/100 Mbps s maximálně 128 (Logix - 64 TCP/IP) zařízeními. Pro připojení využívá měděný vodič s konektorem Ethernet RJ45 kategorie 5.

2.3.5 Modul SERCOS Interface

Modul SERCOS Interface [7] je speciální modul navržený pro řízení servomotorů. Zajišťuje komunikaci mezi PLC a pohonem. Pomocí jednoho modulu lze řídit 3, 8 nebo až 16 os. Samotné propojení pohonu a PLC je dvojicí optických kabelů, které dokáže nahradit až 18 drátových vodičů na osu. Detailní informace o pohonu jsou přenášeny

obousměrně. Pohon může být vzdálen až 200 m od modulu při použití skleněných vláken (32 m při plastových).



Obrázek 2.11: Modul SERCOS Interface

Pro manipulátor byl zvolen dostačující typ modulu 1756-M08SE. Umožňuje ovládat až 8 os současně. Přenosová rychlost je 4 nebo 8 Mbps, od které se odvíjí doba cyklu 0,5 až 2 ms. Propojení s pohonem je pomocí plastových optických kabelů 2090-SCEP3-0 z chlorovaného polyethylenu.

2.4 PanelView Plus 700



Obrázek 2.12: PanelView Plus 700

PanelView jsou zařízení používané k ovládání a vizualizaci technologických procesů. Patří mezi tzv. HMI zařízení. Usnadňují ovládání a kontrolu jednotlivých zařízení, částí výrobních linek i rozsáhlejších procesů.

Pro vizualizaci manipulátoru je použit PanelView Plus 700 [9][10]. Je to grafický barevný displej s klávesnicí. Je navíc dotykový a velmi odolný. Velikost displeje je 6,5 palce. Komunikovat může pomocí sériové linky a přes EtherNet. Rozhraní umožňují velice snadné připojení dalších modulů. Panel má slot pro karty CompactFlash 1 a disponuje paměťovým prostorem rozšiřitelným na 256MB RAM a 512MB CompactFlash. Napájení je 85-264V AC nebo 18-32V DC. Pomocí USB portu lze připojit i klávesnici, myš nebo tiskárnu. Jednoduchým způsobem se dá rozšířit panel o komunikační modul pro DH+/DH-485, DeviceNet nebo ControlNet.

2.5 RS Linx

RS Linx je software umožňující propojení komunikace různých programů od Rockwell Software, které obsluhují různá zařízení. V této práci byl použit na propojení programů RS Logix 5000 a RS View Studio a pro zjištění dostupných zařízení na síti EtherNet. Pomocí stromové architektury je jednoduchý přístup i na jiná zařízení, která nejsou připojena přímo do sítě EtherNet, ale prostřednictvím jiných komunikačních modulů. To bylo využito například pro zjištění stavu měničů, které jsou připojené pomocí optických kabelů a modulu SERCOS Interface. Při chybě lze tak rychle zjistit, zda je problém v komunikaci nebo nastala chyba osy, měniče nebo motoru atd.

V následující tabulce 2.2 je kompletní seznam modulů v použitém PLC. Jsou zde uvedena i zařízení připojená pomocí komunikačních a řídicích modulů.

Tabulka 2.2: Moduly PLC a připojená zařízení

Pozice	Označení modulu	Charakteristika
00	1756-L62S LOGIX5562	Primární procesor
01	1756-LSP LOGIX SAFETY	Safety Partner procesor
02	1756-ENBT/A	EtherNet/IP komunikační modul
03	1756-DNB/C	DeviceNet komunikační modul
04	—	Prázdný slot
05	1756-CNB/E	ControlNet komunikační modul
06	1756-M08SE	Sercos Interface v PLC
– 00	1756-M08SE	Sercos Interface v rozváděči manipulátoru
– 11	2093-ACO5MP5	IAM převodník v rozváděči (osa X)
– 12	2093-AMP5	AM převodník v rozváděči (nevyužitý)
– 13	2093-AMP5	AM převodník v rozváděči (osa Y)
– 14	2093-AMP5	AM převodník v rozváděči (osa Z)

2.6 RS Logix 5000

Program RS Logix 5000 se obecně používá na programování programovatelných automatů. Umožňuje programovat všemi obvyklými jazyky pro PLC, tj. Instruction List (IL), Ladder Diagram (LD), Sequential Function Chart (SFC), Structured Text (ST) a Function Block Diagram (FBD).

V bakalářské práci byl zvolen jako programovací jazyk Ladder diagram. V této kapitole budou nejprve popsány používané proměnné a instrukce. Poté stručně funkce programu a podrobnější popis důležitých částí.

2.6.1 Nastavení

Nejprve byly nastaveny základní vlastnosti. Na kartě General je to Axis Configuration jako Servo. Jako Motion Group byla zvolena vytvořená skupina MotionGroup1. Dále nastavení každého modulu (OsaX, OsaY nebo OsaZ), jeho typu 2093-AMP5 a správné pozice, která byla zjištěna pomocí RS Linx. Lze ji také nalézt v záložce I/O Configuration programu RS Logix. Na kartě Drive/Motor jsou to parametry převodníku a motoru. Na kartě Homing je možné změnit počáteční polohu (Home position). Ta je využívána jako nulová poloha na ose, pro kalibraci výpočtu polohy a pro instrukci Motion Axis Home. Na kartě Tune a Dynamics jsou parametry pohybu, maximální rychlost, zrychlení, zpomalení apod. Na kartě Limits se nastavují limity pro osu.

Pro zajištění bezpečnosti jsou použity oba typy limitů, Hard Travel limity, které jsou dány pozicí fyzických koncových spínačů na ose, a Soft Travel limity nastavitelné na libovolnou pozici. To bylo následně použito pro programové ošetření limitů. Díky tomu se při překročení Soft limitů fyzicky manipulátor nedostane na Hard limity a nemůže nastat kolize. K tomu je nutné ještě nastavit na kartě Fault Actions, co se má provést při dosažení těchto limitů. Pro Soft limity byla tedy nastavena pouze změna statusu (Status Only), což je ošetřeno v samotném programu a způsobí zastavení osy, ale pohon se nevypne. Pro Hard limity je nastaveno zastavení řízení (Stop Motion) a osa se tím zablokuje proti dalšímu pohybu. Pro zpětné uvedení do provozu je třeba odtlačit manipulátor za koncový spínač, aby se odstranila příčina blokování, a také restartovat osu.

Nastavení software limitů:

X: -10 – +540 mm

Y: -10 – +530 mm

Z: -20 – +120 mm

2.6.2 Proměnné

Pro ovládání osy a funkci programu bylo potřeba nadefinovat vlastní proměnné (tags). S označením x (y, z) jsou proměnné používané pro ovládání a nastavení parametrů jednotlivých os, man pro manuální ovládání, MI pro Motion instrukce. Nejprve budou vysvětleny proměnné, které se nevztahují ke konkrétní ose.

Tabulka 2.3: Společné proměnné

Název	Typ	Funkce
initial	bool	pomocná proměnná pro inicializaci
initial_acum	dint	aktuální čas inicializace
initial_pom	bool	příznak inicializace
initial_start	bool	spuštění inicializace
initial_timer	timer	časovač inicializace
group_reset	bool	spuštění restartu os

Proměnná `initial` se nastaví při spuštění programu. Bit `initial_start` spouští program pro inicializaci a `initial_pom` je pomocný příznak právě probíhající inicializace, aby nemohl být spuštěn jiný program. Časovač `initial_timer` běží od spuštění inicializace a v `initial_acum` je jeho aktuální hodnota. Proměnnou `group_reset` se resetuje celá skupina os.

Seznam tagů také obsahuje některé speciální položky. Skupina os `MotionGroup1` je typu `Motion.Group` a představuje vlastnosti této skupiny. Je zde kompletní aktuální stav (`GroupStatus` apod.) celé skupiny os. Daleko obsáhlejší jsou příznaky všech možných chybových stavů, fyzických (`PhysicalAxisFault`), chyby nastavení (`ConfigFault`) atd. Dále je zde `0saX` (`0saY`, `0saZ`) typu `Axis.Servo_Drive`. Obsahuje také několik statusů (`Axis-Status`, `MotionStatus`, `DriveStatus`), které lze využít zejména pro kontrolu stavu systému při složitějších operacích. Důležitější jsou zejména nastavitelné parametry a z pohledu bezpečnosti příznaky chyb. Prostřednictvím parametrů je možné nastavit vlastnosti pohybu (rychlost, zrychlení, zpomalení, maximální točivý moment apod.). Chybové příznaky indikují poruchy nebo jejich příčinu (`OverSpeedFault`, `MotorOvertempFault`, `Overtravel-Fault`).

Proměnné pro konkrétní osy budou vysvětleny pouze pro osu X. Ostatní osy jsou téměř totožné, proto budou zmíněny pouze jejich odlišnosti. Označení SOT symbolizuje proměnné, které byly použity pro vyřešení překročeného software limitu při manuálním ovládání.

Proměnná `x_fault` obsahuje číslo poruchy nebo chyby vzniklé během ovládání. Může nabývat hodnot 0 až 9, přičemž 0 znamená bezchybný stav a ostatní čísla symbolizují chybový stav, který se následně objeví ve vizualizaci. `x_position` a `x_speed` obsahují aktuální stav pohonu a `x_man_accel`, `x_man_deccel`, `x_man_speed` jsou počáteční nebo uživatelem nastavené parametry pohybu. Bity `x_man_gohome`, `x_man_right` a `x_man_left` ovládají pohyb osy doprava, doleva a na home pozici. Pouze pro osu Z jsou zde použity bity `z_man_up` a `z_man_down` aby bylo ovládání srozumitelné. Bitem `x_man_reset` se provede restart a následně smazání chybových stavů.

Tabulka 2.4: Proměnné pro osu

Název	Typ	Funkce
<code>x_fault</code>	int	číslo zobrazené poruchy
<code>x_position</code>	real	aktuální pozice osy
<code>x_speed</code>	real	aktuální rychlost osy
<code>x_man_accel</code>	real	nastavené zrychlení osy
<code>x_man_deccel</code>	real	nastavené zpomalení osy
<code>x_man_speed</code>	real	nastavená rychlost osy
<code>x_man_gohome</code>	bool	pokyn najetí na home pozici
<code>x_man_left</code>	bool	pokyn pohybu doleva
<code>x_man_right</code>	bool	pokyn pohybu doprava
<code>x_man_reset</code>	bool	pokyn resetu osy
<code>x_SOT_NegMemory</code>	bool	negativně překročený limit
<code>x_SOT_PosMemory</code>	bool	pozitivně překročený limit
<code>x_SOT_Ok_To_Override</code>	bool	pokyn návratu za limit
<code>x_SOT_OvertravelResetState</code>	dint	aktuální krok odstranění chyby
<code>x_SOT_SavedConfigFaults</code>	dint	počáteční nastavení limitů
<code>x_SOT_ClearHardBits</code>	dint	limity zakázány
<code>x_SOT_SetSoft</code>	dint	limity znovu nastaveny

V okamžiku, kdy se osa dostane za software limit, tak se uloží příslušná chyba do `x_SOT_PosMemory` nebo `x_SOT_NegMemory` a zachytí se současně systémové nastavení do `x_SOT_SavedConfigFaults` pro pozdější obnovení. Bitem `x_SOT_Ok_To_Override` je nutné potvrdit souhlas se změnou systémového nastavení (vypnutí software limitů) a odjetím mimo limit. Proměnné `x_SOT_ClearHardBits` a `x_SOT_SetSoft` jsou využity pro úpravu nastavených limitů a v `x_SOT_OvertravelResetState` je aktuální krok celé operace.

2.6.3 Instrukce

V programu jsou použity různé instrukce rozdělené do skupin podle typu [5][6]. Bitové instrukce se používají pro monitorování a ovládání stavu bitů. Datové instrukce je upravují nebo přesouvají. Matematické instrukce provádí aritmetické operace a porovnávací instrukce porovnávají hodnoty. Časovače řídí operace závislé na čase. Programovými instrukcemi lze měnit průběh programu (tok logiky, podprogramy). Systémové instrukce čtou nebo zapisují systémová data z nebo do procesoru. Mohou také pracovat s bloky dat pro jiné moduly v jiných sítích rozsáhlejších procesů. Posledním použitým typem jsou Motion instrukce. Jedná se o speciální typ pro ovládání pohonů. Umožňují pohony zapínat, vypínat, zastavovat, nastavovat do definovaných poloh, pohybovat s definovanými parametry pohybu a podobně. Nechybí také instrukce pro řešení chybových stavů, restart jednotlivých os i celých skupin.

Bitové instrukce:

- XIC – (Examine if closed) pokud je proměnná v log. 1, je kontakt průchozí.
- XIO – (Examine if open) pokud je proměnná v log. 0, je kontakt průchozí.
- OTL – (Output latch) nastaví výstup trvale do log. 1.
- OTU – (Output unlatch) nastaví výstup trvale do log. 0.

Datové instrukce:

- MOV – (Move) kopíruje obsah Source do Dest.

Matematické instrukce:

- DIV – (Divide) dělí obsah Source A obsahem Source B a výsledek uloží do Dest.

Porovnávací instrukce:

- EQU – (Equal) testuje rovnost dvou proměnných A a B, při rovnosti je příčka průchozí.
- NEQ – (Not equal) testuje nerovnost dvou proměnných A a B, při nerovnosti je příčka průchozí.
- LES – (Less than(A;B)) testuje jestli A je menší než B, pokud ano je příčka průchozí.
- GRT – (Greater than(A;B)) testuje jestli A je větší než B, pokud ano je příčka průchozí.

Časovače :

- TON – (Timer on delay) časovač zpožděného sepnutí. Měří čas od aktivace čítače (aktivní přička, bit .EN) po doběhnutí nebo deaktivaci. Bit .PRE obsahuje požadované zpoždění v sekundách, aktuální hodnota je v .ACC, po uběhnutí času se nastaví bit .DN.
- RES – (Reset) Reset čítače nebo časovače.

Programové řídicí instrukce:

- JSR – (Jump to subroutine) skok do podprogramu s nastaveným názvem.

Systémové (vstupní/výstupní) instrukce:

- GSV – (Get system value) načte specifická systémová data a uloží je do Dest. Nastavuje se zde třída objektu, jméno objektu a jméno atributu, který se načítá.
- SSV – (Set system value) nastaví specifická systémová data podle zdroje.

Motion instrukce (speciální instrukce pro ovládání pohonů):

- MSO – (Motion servo on) instrukce aktivuje osu.
- MSF – (Motion servo off) instrukce zruší všechny probíhající instrukce řízení a přepne osu do pohotovostního režimu.
- MASR – (Move axis shutdown reset) instrukce nastaví osu do stavu Ready. Všechny chyby osy se automaticky mažou.
- MAFR – (Move axis fault reset) instrukce smaže všechny chyby osy. Pokud ale nejsou před spuštěním instrukce odstraněny všechny příčiny chyb, zůstanou chybové příznaky nastaveny.
- MAH – (Motion axis home) najede s osou do home pozice, tím se i zkalibruje údaj o poloze.
- MAJ – (Motion axis jog) pohybuje osou se zadanými parametry (rychlost, zrychlení, zpomalení) na zvolenou stranu.
- MAS – (Motion axis stop) instrukce zastaví osu s nastaveným zpomalením.
- MGSR – (Motion group shutdown reset) instrukce nastaví skupinu os do stavu Ready. Automaticky mažou chyby všech os ve skupině.

2.6.4 Program

Program je kompletně vytvořený v Ladder diagramu. Skládá se ze tří základních částí. MainRoutine je hlavní program, který musí být v každém projektu a je automaticky vytvořen při založení nového projektu. Ovládá zde spouštění ostatních podprogramů a vyhodnocuje vzniklé chyby v systému. Podprogram Initial nastavuje celý manipulátor do základního stavu. V tomto stavu je připraven podprogram Manual vykonávat příkazy obsluhy na pohyb ve všech třech osách.

Kompletní program v Ladder diagramu je na příloženém CD, které obsahuje funkční projekt i zhotovený report ve formátu PDF. Bloková struktura programu je na obrázku 2.13.

MainRoutine

Základní program MainRoutine při spuštění nastaví bit `initial`. Zároveň s ním se zablokuje spuštění programu pro ovládání, protože zatím neproběhla inicializace os. Také se spustí časovač `initial_timer`, aby byla k dispozici informace o době konfigurace pohonů, které se konfiguruje také se spuštěním programu. Pokud jsou pohony nakonfigurovány a nehlásí žádnou chybu, je zpřístupněno spuštění podprogramu pro inicializaci os sepnutím bitu `initial_start`.

Po úspěšné inicializaci os je manipulátor nastaven do počáteční pozice [0,0,0]. Resetováním bitů `initial` a `initial_pom` je umožněno ovládání pohonů programem Manual. I chod tohoto programu je podmíněn bezchybným stavem všech os.

Součástí MainRoutine jsou i příčky 5-11, které zajišťují reset os. Bitem `x_man_reset` (y, z) se spustí instrukce `x_MI_MASR` a následně `x_MI_MAFR`. Pro reset všech os ve skupině je k dispozici `group_reset` a `group_MI_MGSR`.

Příčky 12-21 resp. 12-41 řeší již zmíněný problém s překročením software limitů. Po zjištění zřekročeného limitu příznakem `OsaX.PosSoftOvertravelFault` (nebo `Neg`) je osa zastavena instrukcí `MAS1` a pomocí `GSV` se uloží obsah `FaultConfigurationBits` atributu do `x_SOT_SavedConfigFaults`. Následně se uloží směr překročení (`x_SOT_PosMemory` nebo `Neg`) a program vyčkává potvrzení dalšího postupu. Po potvrzení se resetuje bit `x_SOT_ClearHardBits.0` a instrukcí `SSV` se přepíše nastavení osy. Stávající překročený limit již neplatí, tudíž stačí restart osy a tím i vymazání chybového stavu. S osou je tedy možné pohybovat. Proto je podle předem uloženého bitu s charakterem překročení (`Pos-NegMemory`) zavolána instrukce `x_SOT_MAJ_Off_SOT`. Osa tedy odjede za limit (pro jistotu o 5mm dále), zastaví se a instrukcí `SSV` jsou znovu zapnuty software limity. Poté je možné pracovat s osou znovu v plném rozsahu.

Zbylé 3 příčky vyhodnocují vzniklé chyby na ose. Podle příznaku chyby se do proměnné `x_fault` uloží číslo příslušné chyby. Pro samotný program má pouze význam přerušení chodu neznámou chybou. Konkrétní informace o chybě je využita ve vizualizaci, kde se na operátorský panel vypíše stručný popis chyby.

Initial

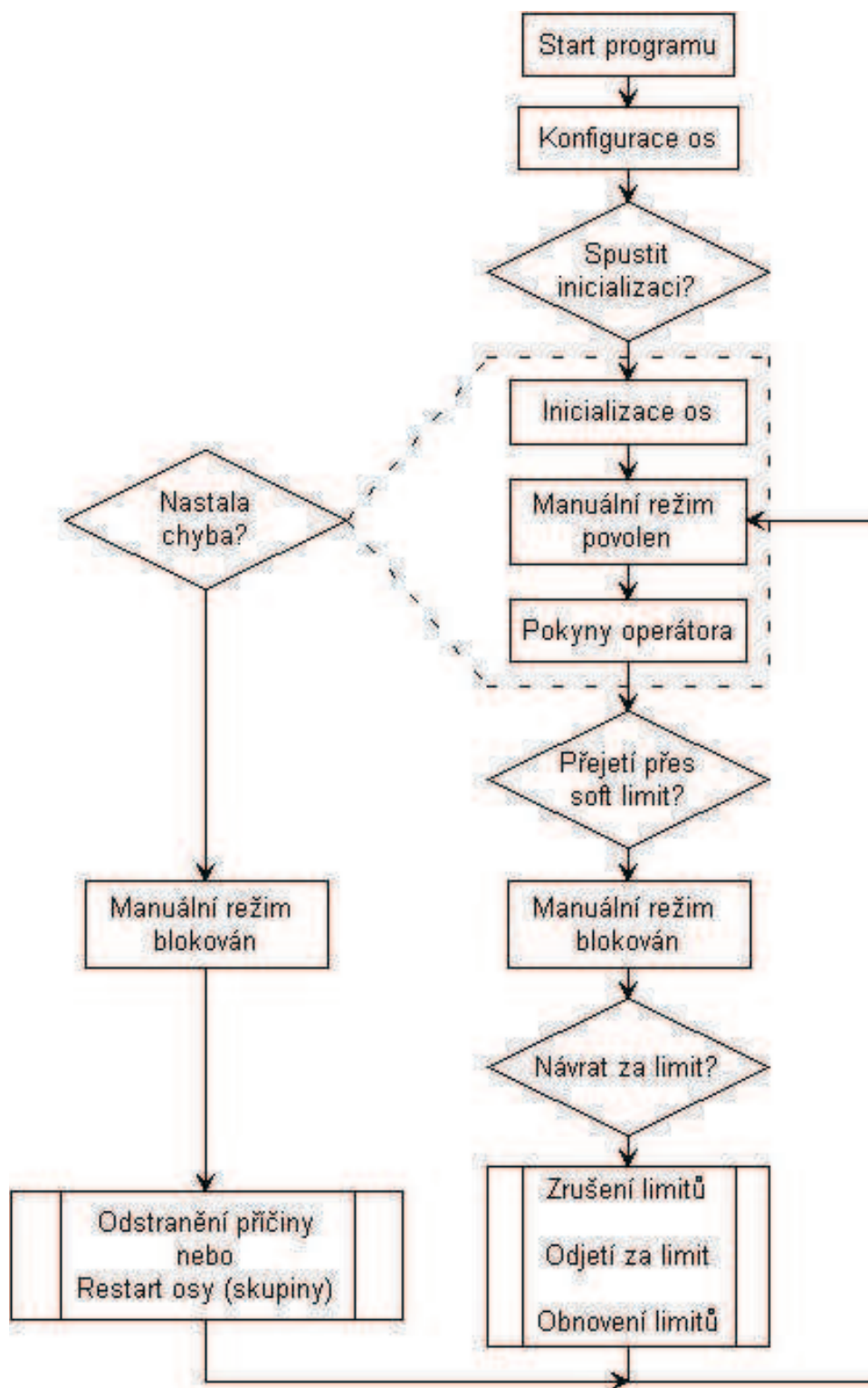
Program Initial je spuštěn před samotnou manipulací s manipulátorem. Ihned po zavolání podprogramu je každá osa restartována pro případ nějaké chyby (MASR1). Na počátku jsou také nastaveny počáteční parametry pro pohyb (`x_man_accel`, `deccel` a `speed`), které lze poté kdykoliv změnit. Pohon je zapnut do stavu Ready (`x_MI_MS0_1`) a poslán na home pozici (`x_MI_MAH_1`). Po dosažení home pozice je zkalibrována aktuální poloha a osa je vypnuta (`x_MI_MSF_1`). Tím je program pro inicializaci ukončen a manipulátor je připraven vykonávat povely obsluhy.

Manual

Program Manual následuje po dokončení inicializace. Vykonává povely posunu ve zvoleném směru a najetí do home pozice.

Bitem `x_man_gohome` se zapne pohon (`x_MI_MS0_2`) a osa se přesune (`x_MI_MAH_2`) do home pozice. Osa se nepohybuje zvolenou rychlostí, ale předdefinovanou v Axis Properties na kartě Homing.

Bity `x_man_right` a `x_man_left` (`z_man_up`, `down` pro osu Z) nejprve zapnou pohon, pokud již není zapnutý, a poté pohybují s osou (`x_MI_MAJ_1` nebo `2`) požadovaným směrem. V tomto případě je možné rychlost, zrychlení a zpomalení zvolit (`x_man_speed`, `accel` a `deccel`). Zastavení pohybu je svázáno s instrukcí `x_MI_MAS`.

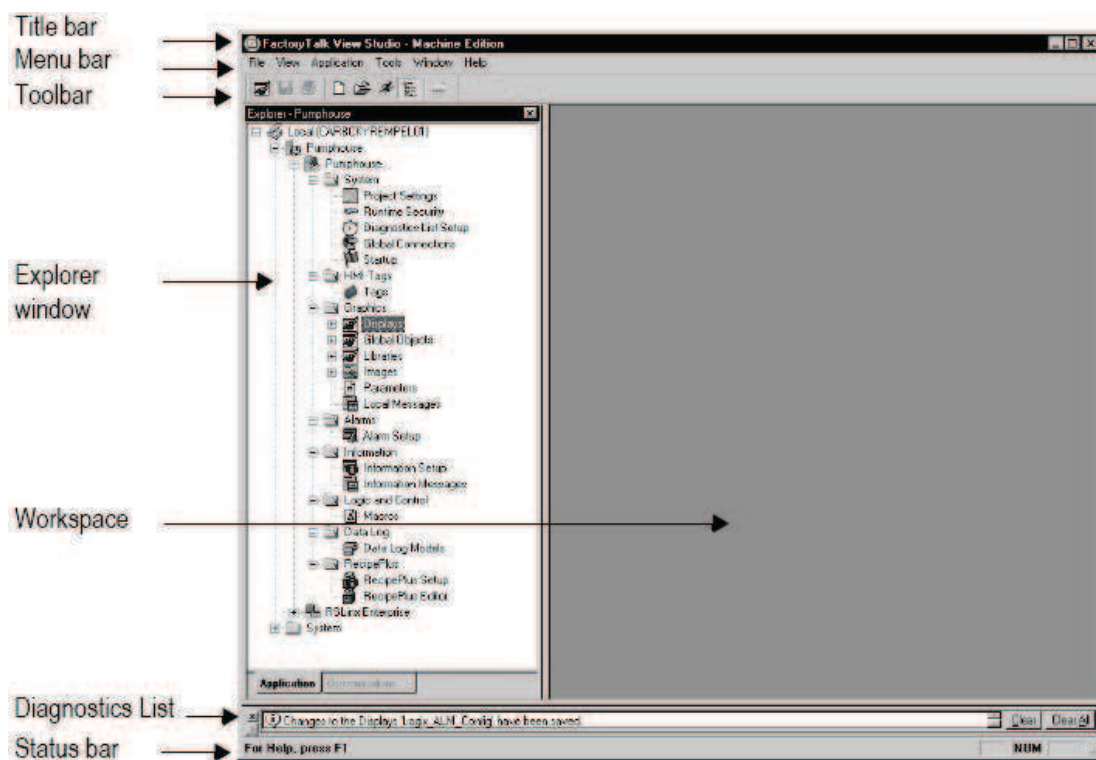


Obrázek 2.13: Vývojový diagram programu

2.7 RS View Studio (Factory Talk View Studio)

Program RS View Studio (nový název je Factory Talk View Studio) se používá zejména pro vizualizaci např. technologických procesů. Poměrně často se používá s produkty Logix. Je schopen pracovat i s širokou škálou produktů jiných výrobců.

Prostředí samotného programu je velice přehledné. V horní části je klasický panel a rolovací menu a panel nástrojů se symboly pro přímou volbu funkcí, což velice urychluje práci. Vlevo je pak Explorer projektu, kde je stromově zobrazena celková struktura projektu, od systémových nastavení přes grafickou část až po fyzickou konfiguraci používaných zařízení. Vpravo je pak Workspace pro vytváření vizualizace. V dolní části je pak Diagnostics List, kde se zobrazují probíhající systémové aktivity. Status bar pod ním zobrazuje informace např. o aktivovaném objektu, zvoleném nástroji nebo položce v menu.



Obrázek 2.14: Prostředí RS View Studia [11]

2.7.1 Nastavení

Před samotným začátkem vytváření vizualizace bylo nutné nastavit základní parametry projektu. Nejprve bylo v záložce Project Settings na kartě General nastaveno Project window size na 640x480. Toto rozlišení bylo dáno vlastnostmi použitého displeje Panel

View Plus 700, aby při kopírování výsledné runtime aplikace do displeje nenastal problém se správným zobrazením. Pro správné zkopírování runtime aplikace bylo ještě třeba nastavit v záložce Startup položku Initial Graphic a vybrat počáteční displej zobrazení. Jednotlivé displeje byly přidány v záložce Graphics a Displays vložením a pojmenováním nové položky.

Dále bylo nastaveno hlavní zařízení, jehož funkce a ovládání bude vizualizováno. To lze provést v záložce propojené přímo s RS Linx, konkrétně v podsložce Communication Setup. Podle přidělené IP adresy byl v záložce Target vybrán příslušný automat, vytvořeno a pojmenováno nové zařízení a nakopírována konfiguraci automatu do záložky Local. Tím bylo docíleno toho, že při přiřazování proměnných do objektů je možné používat i proměnné vytvořené pro automat v programu RS Logix 5000.

2.7.2 Objekty

Pro přehledné zobrazení stavu a ovládání manipulátoru byly použity různé druhy objektů [11][12], jejichž funkce bude stručně popsána v této podkapitole.

- Display – základní grafický objekt, na kterém se vytváří výsledná vizualizace. Je možné vytvořit několik objektů tohoto typu a pomocí speciálních tlačítek se mezi nimi přepínat.
- Line – běžná čára s nastavitelnou šířkou a barvou.
- Panel – obdélník s nastavitelným okrajem a barvou.
- Text – slouží k zobrazení vloženého textu a nastavení jeho vlastností.
- Shutdown button – tlačítko pro ukončení celé aplikace.
- Goto display button – tlačítko pro přepnutí na nastavený displej.
- Return to display button – tlačítko pro návrat na předchozí displej.
- Momentary push button – tlačítko stiskem změní proměnnou na nastavenou hodnotu (log. 1 nebo 0), po uvolnění tlačítka se proměnná vrací do původního stavu.
- Maintained push button – přepínač. Při stisku změní hodnotu nastavené proměnné na jednu hodnotu, při dalším stisku ji změní na druhou hodnotu.
- Multistate indicator – indikuje hodnotu proměnné nadefinovanými stavy. Podle stavu proměnné je možné měnit barvu, vzhled a popisek.
- Numeric display – zobrazuje číselnou hodnotu proměnné ve zvoleném tvaru se zvolenou přesností.

- Numeric input enable button – používá se pro vkládání a změnu hodnoty proměnné za běhu aplikace, např. pro změnu parametrů. Pro vkládání je možné nastavit meze zadávané hodnoty. Zároveň je možné nastavit do popisku tuto proměnnou a aktuální hodnotu již nemusíme zobrazovat pomocí např. Numeric displeje.
- Bar graph – zobrazuje aktuální hodnotu proměnné ve formě bar grafu. Umožňuje nastavit zobrazení hodnoty nebo procenta.
- Trend – vykresluje historii nebo aktuální číselnou hodnotu proměnné. Je možné zobrazovat více proměnným v závislosti na čase (zobrazuje se pouze nastavený časový úsek, po jeho překročení nelze starší hodnoty zpětně zjistit) nebo dvě vůči sobě (XY zobrazení).

2.7.3 Vizualizace

Před samotným začátkem vytváření vizualizace byl přenastaven rastr pro sazbu objektů. To lze provést v záložce Grid Settings.

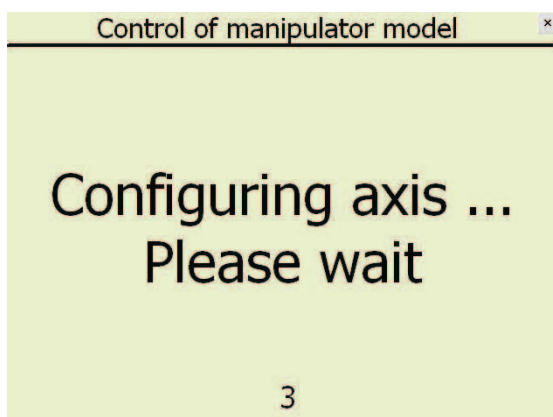
Vizualizace se skládá ze základních objektů typu Display. Jeden zobrazuje samotné ovládání a aktuální informace o manipulátoru. Druhý je pomocný a slouží pro zobrazení nebo změnu parametrů pohybu jednotlivých os. Oba displeje mají společných několik objektů. Je to horní nápis typu Text (Control of manipulator model), černá čára pro oddělení nadpisu od ostatních objektů a Shutdown tlačítko pro ukončení aplikace. Toto tlačítko slouží na vypnutí aplikace a je svázané s tlačítkem K7 na panelu. Bez něj by nebylo možné dostat se do vlastního menu panelu, ve kterém se spouští zvolená aplikace, mění nastavení panelu apod.

Control

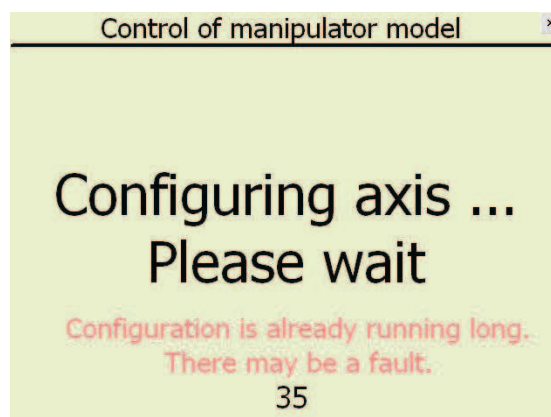
Tento displej je možné považovat za hlavní část vizualizace. Obsahuje velké množství objektů a informací a jejich popis bude nejlépe srozumitelný v závislosti na jednotlivých fázích probíhajícího programu. Pro tento displej byla vytvořena proměnná `pos_hist`, která je zároveň jedinou proměnnou v programu RS View. Zajišťuje zde viditelnost vybraných objektů při volbě zobrazení historie poloh os. Samotná viditelnost byla nastavena u všech objektů tohoto displeje. Bylo nutné vázat ji na stav klíčových proměnných, aby vizualizace měnila vzhled podle aktuálního stavu manipulátoru. Zobrazení (viditelnost) se nastavuje jednotlivě u každého objektu přes kontextové menu Animation ⇒ Visibility.

Po spuštění programu se začnou konfigurovat osy manipulátoru. Dokud nejsou osy nakonfigurovány, je zobrazena informace o konfiguraci 2.15 a čas od spuštění programu. Nápis je běžný objekt Text. Čas je načítán jako aktuální hodnota čítače `initial_timer`

z proměnné `initial_acum` a zobrazen prostřednictvím Numeric displeje. Během vývoje programu a testování bylo zjištěno, že osy se nakonfigurují přibližně za 15 sekund (výjimečně do 20 s). Proto se po překročení 30 s zobrazí nápis 2.16, upozorňující na možné potíže při konfiguraci, aby obsluha věděla, že neprobíhá standardně.

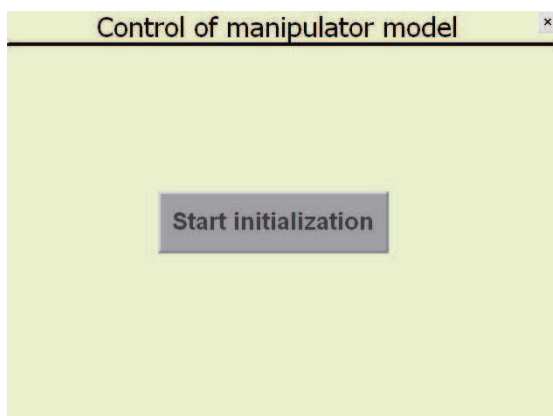


Obrázek 2.15: Konfigurace os

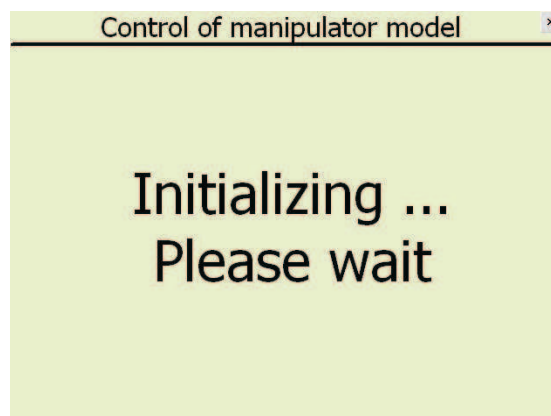


Obrázek 2.16: Konfigurace os - chyba

Po úspěšné konfiguraci se zobrazí tlačítko typu Momentary push button pro start inicializace 2.17. Tlačítko je propojené s proměnnou `initial_start` a v případě, že nastala nějaká chyba, restartuje čítač a spouští podprogram pro inicializaci Initial. Zobrazí se tedy informace o probíhající inicializaci 2.18.



Obrázek 2.17: Start inicializace



Obrázek 2.18: Probíhající inicializace

Po inicializaci se zobrazí kompletní ovládání manipulátoru. Na obrázku 2.19 je zobrazeno za normálního stavu bez chyb a v nulových polohách po inicializaci. V horní části nalevo je tlačítko Position history typu Maintained push button pro zobrazení historie poloh všech tří os. Tlačítko ovládá proměnnou `pos_hist` a je navíc propojeno s tlačítkem

K1 na klávesnici panelu. Pod ním je tlačítko Set speed typu Goto display button, které přepíná zobrazení na displej Setting a je propojeno s tlačítkem K2 na panelu.

V pravé horní části se zobrazují aktuální údaje. Pro zvýraznění jsou oddělena pomocí objektů Panel s nastavenou barvou a vzhledem okraje. Popisy jsou objekty Text. Číselně prostřednictvím objektů Numeric display se vypisují aktuální polohy a rychlosti os. Objekty jsou napojeny na proměnné `x_position` a `x_speed` (podobně pro `y` a `z`). Položka Axis status je typu Multistate indicator. Informuje obsluhu o stavu jednotlivých pohonů. Je propojena s proměnnou `x_fault` a podle její hodnoty mění popis stavu. Obsahuje devět stavů, které odpovídají vzniklým chybám. Tyto chyby se vyhodnocují v Ladder diagramu v části MainRoutine.

Tabulka 2.5: Stavy osy v Multistate indicator

Číslo	Popis
0	Normal
1	Axis fault
2	Module fault
3	Drive fault
4	Hard Overtravel fault
5	Soft Overtravel fault
6	Motor Overtemp fault
7	Drive Overtemp fault
8	Sercos fault
9	Feedback fault

V dolní části vlevo jsou ovládací prvky. Pro pohyb os jsou použita běžná tlačítka typu Momentary push button, která mají pouze v položce Caption namísto textu napsány šipky. Ovládají bity `x_man_right` a `x_man_left` (up a down pro osu Z). Po obou stranách tlačítek jsou umístěny Multistate indicator objekty, které zobrazují stav krajních limitů os. Obsahují tři stavy. State0 (zelený, hodnota 0) znamená bezchybný stav. Při překročení software limitu nastane State1 (oranžový, hodnota 1). Při překročení i hardware limitu nastane State2 (červený, hodnota 2). Aktivace příslušných stavů je realizována pomocí Expression Editoru. V něm je možné vytvořit např. podmínky a různé druhy operátorů a pomocí nich vyhodnocovat stav objektu. Pro příklad v levém indikátoru osy X je tato podmínka:

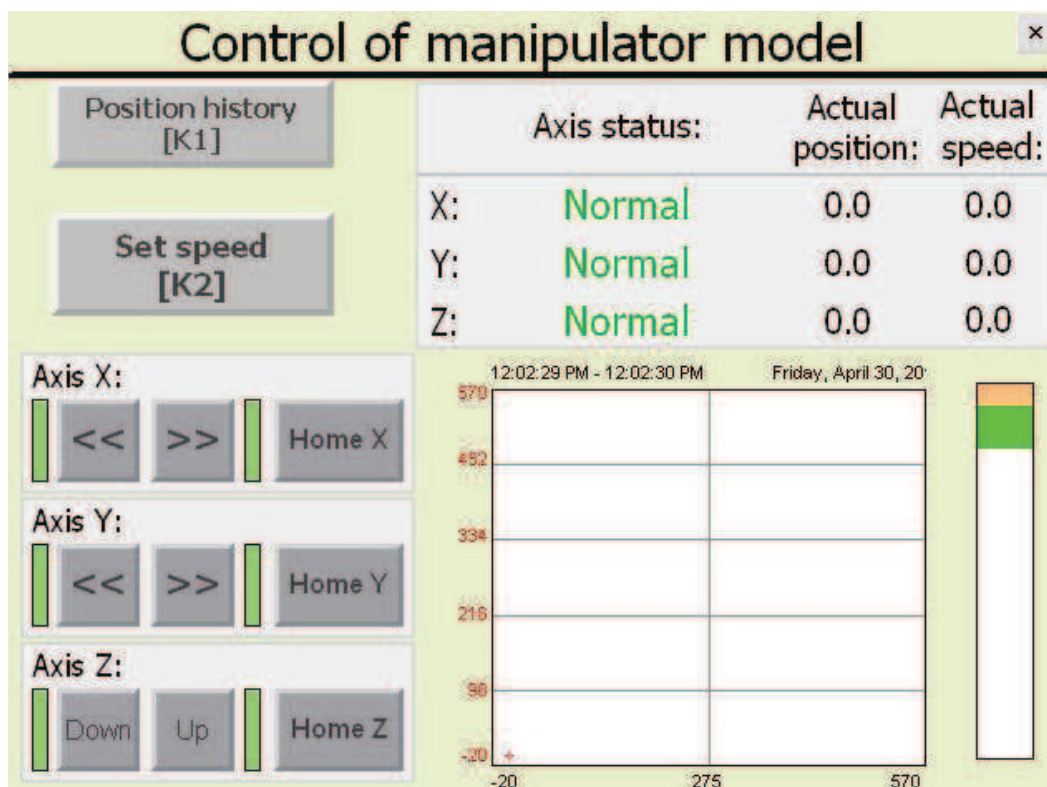
```
If {[New2]OsaX.PosHardOvertravelFault} == 1
Then 2
Else If {[New2]OsaX.PosSoftOvertravelFault} == 1
Then 1
Else 0
```

Podmínky využívají přímo interní příznaky osy. V tomto případě se kontroluje nejprve chyba pozitivního překročení hardware limitu, pokud je překročen nastane State2. Pokud překročen není, tak se kontroluje pozitivní překročení software limitu. Pokud je překročen nastane State1, pokud ne State0. Tyto stavy se samozřejmě projeví i změnou položky Axis status. V této části jsou ještě u každé osy tlačítka Home X (Y, Z). Pomocí nich jsou ovládány bity `x_man_gohome` (y, z) a spouští instrukci pro najetí na home polohu.

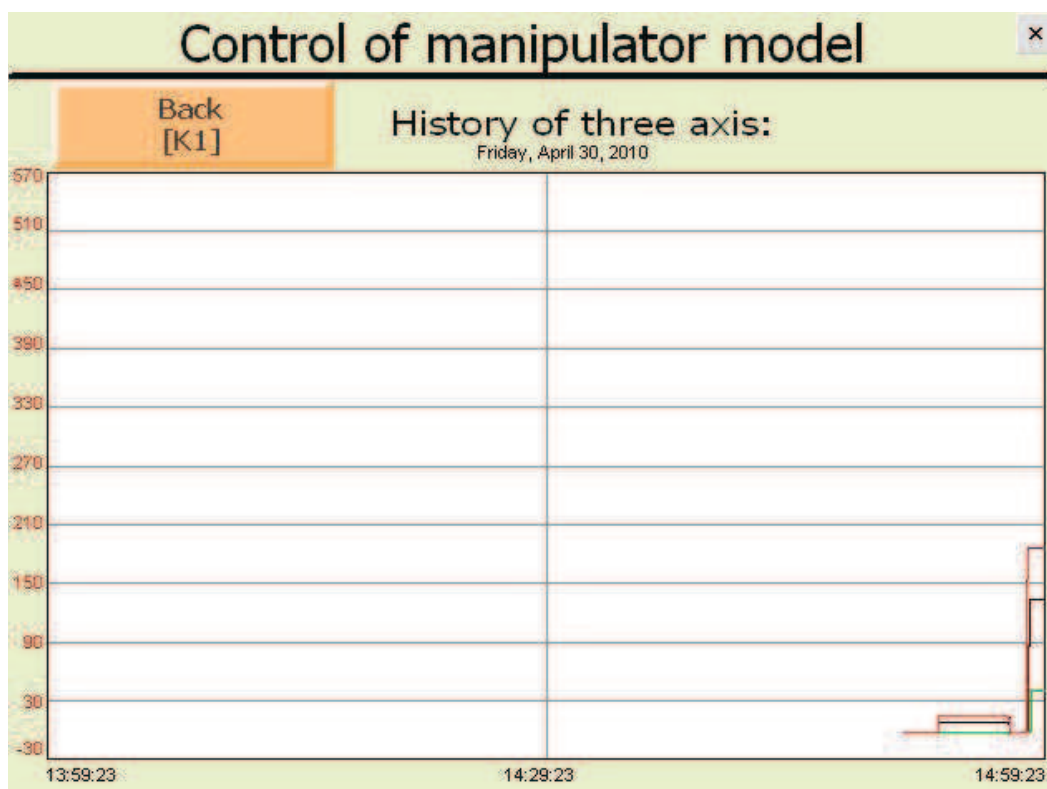
Pravá dolní část je využita pro grafické zobrazení aktuální polohy os. Je zde objekt Trend. Je nastaven pro zobrazení v režimu XY. Toto zobrazení vykresluje vzájemnou polohu dvou proměnných `x_position` a `y_position` vůči sobě. Pro optimální zobrazení byly nastaveny rozsahy os s ohledem i na překročení obou typů limitů. Poloha je zobrazena červeným bodem. Vzhledem k velikosti panelu a celkové přehlednosti nebylo vhodné nastavení menšího rastru. Jedná se tedy o přibližnou polohu na ploše XY a přesné nastavení polohy lze zaručit číselným údajem výše. Vedle objektu Trend je Bar graph. Zobrazuje obsah proměnné `z_position` ve formě sloupcového grafu. Graf se zaplňuje směrem dolů. Slouží pro přibližné zobrazení polohy. Pro informaci o vzdálenosti od software limitu je graf zabarven v horní části oranžovou barvou. Po překročení spodního software limitu se část za limitem také zobrazí oranžovou barvou. Rozsah zobrazení byl zvolen přibližně na úrovni hardware limitů.

Obrázek 2.20 ukazuje stav vizualizace po přepnutí na zobrazení historie os tlačítkem Position history. Pro vykreslení historie je zde také objekt typu Trend. Je ale nastaven na zobrazení proměnných v závislosti na čase. Rozsah pro osu Y je zvolen podle maximálních rozsahů os. Vykreslují se proměnné `x_position`, `y_position` a `z_position`. Pro časovou osu byl nastaven rozsah 1 hodina.

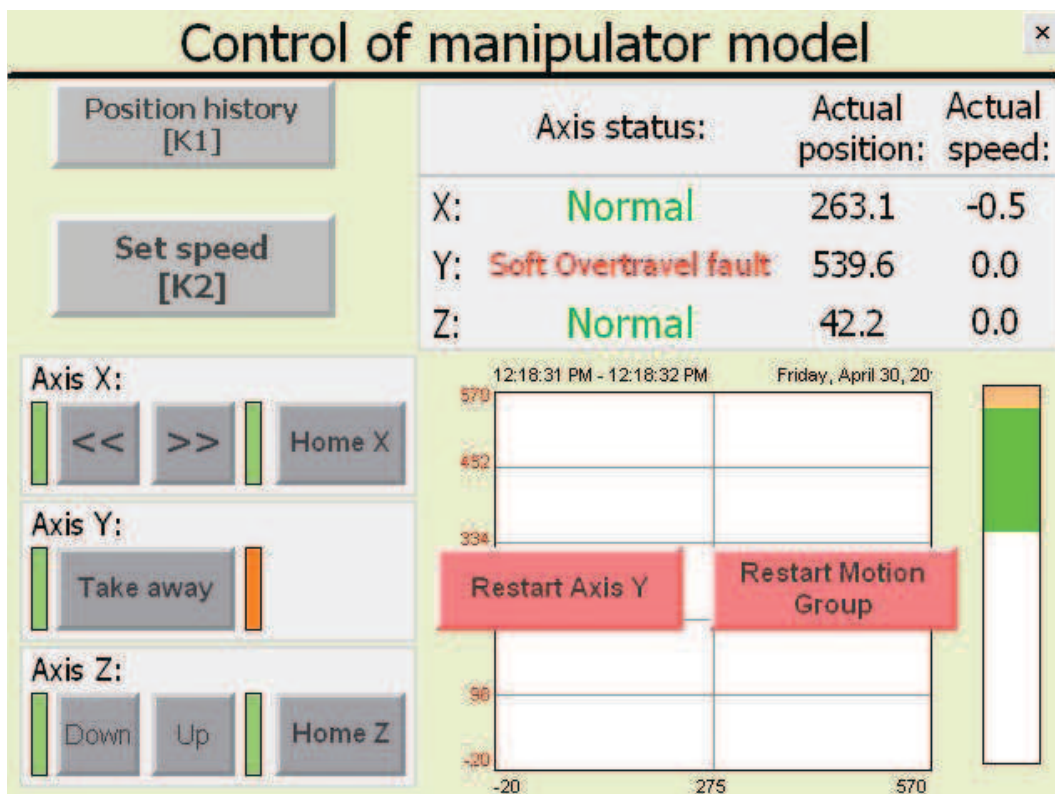
Na obrázku 2.21 je vizualizace při překročení software limitu. V horní části se změnil stav osy Y a informuje o vzniklém problému. Po vzniku tohoto problému ihned zmizela tlačítka pro ovládání pohybu osy a indikátor stavu limitů osy se zbarvil oranžovou barvou. Zpřístupnilo se tlačítko Take away typu Momentary push button, které spíná bit `x_SOT_0k_To_Override` a povoluje zásah programu do bezpečnostního nastavení pohonu. Zároveň se také zviditelnila tlačítka Restart Axis Y a Restart Motion Group ovládající bity `y_man_reset` a `group_reset`. Tato tlačítka jsou tu pro případ, že by zároveň nastalo více chyb a odjetí za software limit by neodstranilo chybový stav osy.



Obrázek 2.19: Ovládání manipulátoru - stav po inicializaci



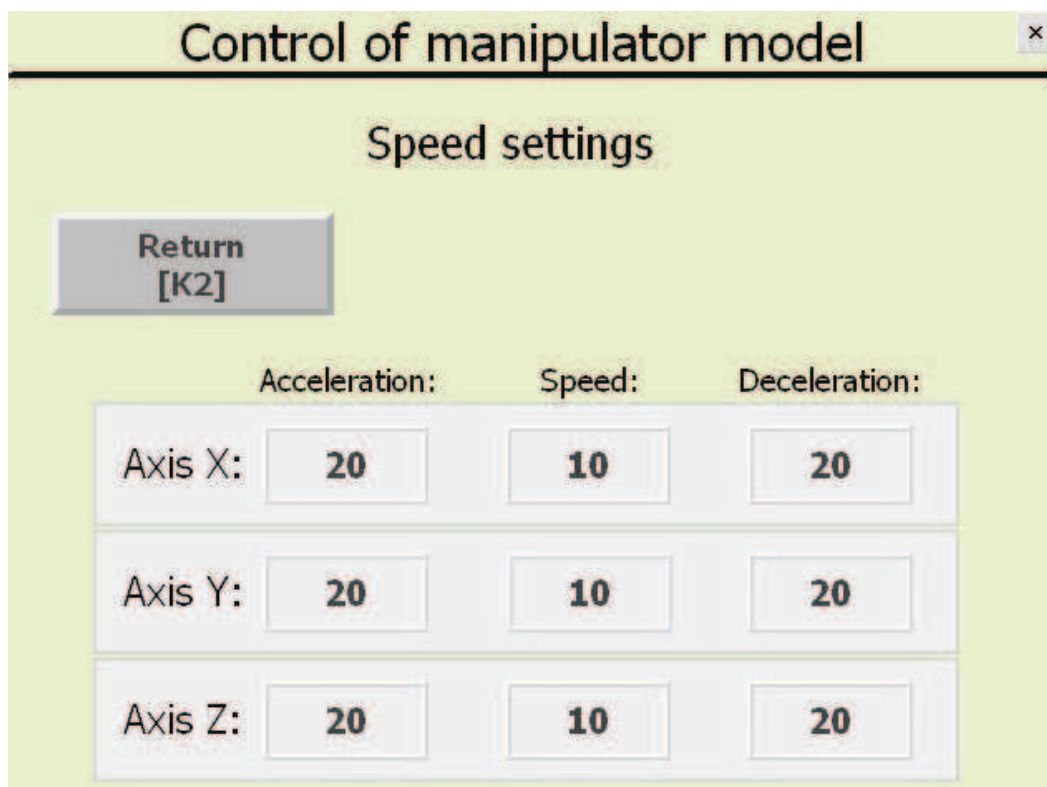
Obrázek 2.20: Historie poloh os



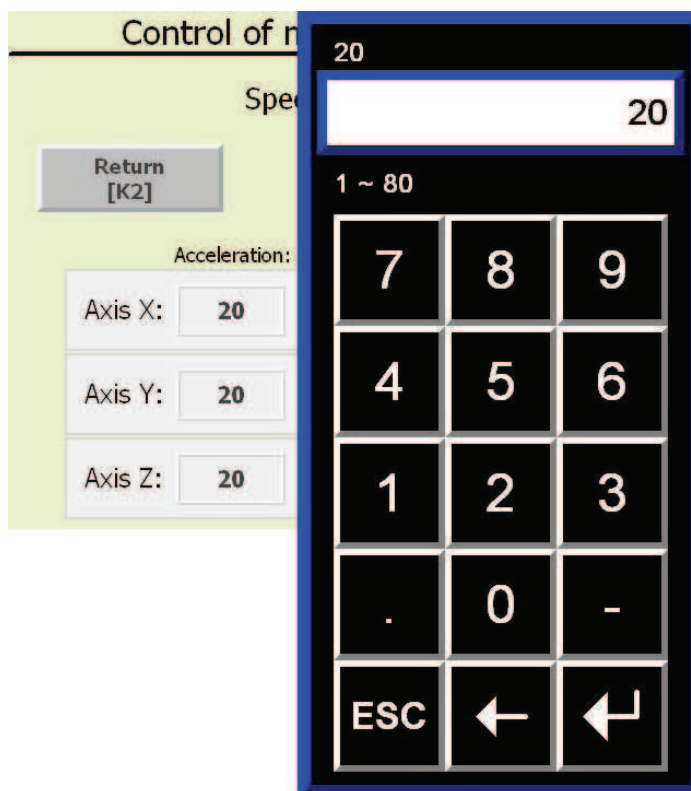
Obrázek 2.21: Překročení software limitu

Setting

Displej Setting 2.22 byl vytvořen pro přehledné nastavení parametrů pohybu manipulátoru. Obsahuje jedno tlačítko Return typu Return to display button spojené s tlačítkem K2 na panelu. Dále jsou zde popisky typu Text a obdélníky typu Panel pro přehledné uspořádání displeje. Hlavní část tvoří devět objektů typu Numeric input enable button. V nastavení mají v položce Value i Caption vždy stejnou proměnnou `x_man_accel`, `x_man_speed` nebo `x_man_deccel` (y nebo z). Není tedy potřeba dalších objektů pro zobrazení aktuálně nastavených hodnot, protože se zobrazují jako popisek objektu. Při kliknutí na tento objekt se zobrazí numerická klávesnice 2.23 pro vkládání hodnot. V horní části je zobrazena aktuální hodnota proměnné, pod ní je navolená hodnota. Nad numerickou klávesnicí jsou meze volené hodnoty (pokud jsou nastaveny) a klávesnice nepovolí vložit hodnotu mimo ně.



Obrázek 2.22: Displej Setting



Obrázek 2.23: Vkládání hodnot

3 ZÁVĚR

Prvním úkolem bylo seznámení s instrumentací firmy RA. To probíhalo zejména před započítím práce a následně během ní podle potřeb řešené problematiky. Základem bylo nastudování techniky ovládání servomotorů pomocí systému Kinetix 2000, který byl navržen právě na řízení servomotorů. Poté následovalo samotné programování automatu třídy GuardLogix, což je provedení automatu ControlLogix určené pro bezpečné aplikace. Jako programovací jazyk byl zvolen Ladder Diagram, kde byly využity speciální instrukce Motion řízení pro servomotory.

Hlavním cílem projektu bylo vytvoření vizualizace manuálního ovládání manipulátoru. To bylo vytvořeno v programu RS View Studio. Nejprve bylo nastaveno prostředí pro samotnou tvorbu. Poté následovalo seznámení s dostupnými objekty v programu a výběr těch nejvhodnějších pro návrh vizualizace. Byl vytvořen výsledný vzhled pro manuální ovládání a volbu hodnot a nastaveny použité objekty s ohledem na výslednou funkci. Výsledná vizualizace byla propojena s programem v PLC a otestována její funkčnost. Testování probíhalo na zmíněném panelu.

REFERENCE

- [1] RA *1756 GuardLogixTM Integrated Safety Controller*
[online], November 2008, 1756-PP009B-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/pp/1756-pp009_-en-p.pdf
- [2] RA *GuardLogix Controller Systems – Safety Reference Manual*
[online], July 2008, 1756-RM093E-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm093_-en-p.pdf
- [3] RA *1756 ControlLogix Controllers Specifications – Technical Data*
[online], May 2009, 1756-TD001A-EN-E, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td001_-en-e.pdf
- [4] RA *GuardLogix Controllers – Uses Manual*
[online], July 2008, 1756-UM020D-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1756-um020_-en-p.pdf
- [5] RA *Logix5000 Controllers General Instructions*
[online], October 2009, 1756-RM003L-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm003_-en-p.pdf
- [6] RA *Logix5000 Controllers Motion Instructions – Reference Manual*
[online], July 2008, 1756-RM007J-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm007_-en-p.pdf
- [7] RA *1756 ControlLogix Integrated Motion Specifications – Technical Data*
[online], May 2009, 1756-TD004A-EN-E, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td004_-en-e.pdf
- [8] RA *ControlLogix Selection Guide*
[online], May 2009, 1756-SG001M-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/sg/1756-sg001_-en-p.pdf

- [9] RA *2711P and 2711PC PanelView Plus Terminals – Product Information*
[online], June 2009, 2711P-PC001A-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/pc/2711p-pc001_-en-p.pdf
- [10] RA *PanelView Plus Terminals – Uses Manual*
[online], November 2009, 2711P-UM001J-EN-P, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/um/2711p-um001_-en-p.pdf
- [11] RA *Factory Talk View Machine Edition – User's Guide Volume 1*
[online], August 2007, VIEWME-UM004E-EN-E, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/um/viewme-um004_-en-e.pdf
- [12] RA *Factory Talk View Machine Edition – User's Guide Volume 2*
[online], August 2007, VIEWME-UM005E-EN-E, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/um/viewme-um005_-en-e.pdf
- [13] RA *Servopohony Kinetix*
[online], září 2007, MOTION-BR005A-CS-P, [cit. 2010-05-20]
http://www.rockwellautomation.cz/applications/gs/EMEA/GSCZ.nfs/files/motion-br005_-cs-p.pdf
- [14] RA *1756 ControlLogix Communication Modules Specifications*
[online], May 2009, 1756-TD003A-EN-E, [cit. 2010-05-20]
http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1756-td003_-en-e.pdf
- [15] REM-Technik s.r.o. *1756 ControlLogix Communication Modules Specifications*
[online], 02810202A0803, [cit. 2010-05-20]
http://www.rem-technik.cz/produkte/pulsotronic/pdf/Inductive_Sensors.pdf
- [16] RA *Essential Components – Katalog*
[online], březen 2008, EC-GL-CA005A-CS-P, [cit. 2010-05-20]
<http://www.rockwellautomation.it/applications/gs/EMEA/GSCZ.nsf/pages/Prumyslovareseni>
- [17] Rockwell Software Inc. *RS Linx*
[Počítačový program], Rev. 2.55.00.16

- [18] Rockwell Software Inc. *RS Logix 5000*
[Počítačový program], Ver. 17.01.00
- [19] Rockwell Software Inc. *FactoryTalk View Studio*
[Počítačový program], Ver. 5.10.00

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A-B Allen - Bradley

AC Alternate Current – Střídavý proud

AM Axis Module

EN Evropská Norma

FBD Function Block Diagram

HMI Human - Machine Interface

IAM Integrated Axis Module

IL Instruction List

IP Intrese Protection – Krytí

LD Ladder Diagram

PLC Programmable Logic Controller – Programovatelný automat

RA Rockwell Automation

RS Rockwell Software

SERCOS SERIAL Realtime COMMunication System

SFC Sequential Function Chart

SIL Safety Integrity Level

ST Structured Text

USB Universal Serial Bus

SEZNAM PŘÍLOH

A Obsah CD s přílohami

45

A OBSAH CD S PŘÍLOHAMI

Příložené CD obsahuje následující soubory:

1. BP_Chlad_Petr.pdf
2. BP_priloha.zip
 - (a) FactoryTalkView_ApplicationDocumenter.pdf – report vizualizace generovaný Application Managerem
 - (b) manipulator.ACD – program v RSLogix 5000
 - (c) manipulator.apa – záloha vizualizace v programu FactoryTalk View Studio
 - (d) RSLogix_5000_Report.pdf – report programu v RSLogix 5000